

# Inlämningsuppgift ASP .NET Core

Täckta kursplansmål:

1. Redogöra för webbramverket ASP.NET Core och dess användningsområde
2. Beskriva hur en bygger webbapplikationer med ASP.NET Razor Pages och MVC

## Övergripande

I denna inlämningsuppgift ska du bygga en webbapplikation i ASP .NET Core Razor Pages. Uppgiften går ut på att bygga en förenklad applikation där man via en webbsida kan se olika roliga event som kommer att ske. Med events menas här t.ex. en konsert på en arena, ett party på en klubb, med mera.

För enkelhetens skull behöver du inte bygga ett administrationsgränssnitt i denna inlämningsuppgift, alltså: du behöver **inte** bygga en extra webbsida där de som håller i eventen ska kunna lägga upp nya event, editera event osv. Istället får du själv definiera ett antal event i förväg och seeda en databas.

Du behöver **inte** heller sköta registrering/inloggning. Om du behöver en användare för att registrera deltagande till ett event så kan du bara plocka första bästa användare från databasen. Resultatet av denna inlämningsuppgift kommer att användas för att öva på registrering och inloggning senare i kursen.

Du bygger hela webbapplikationen som ett projekt i Visual studio eller VS Code, där samtliga delar som utgör applikationen kommer att finnas (både backend och frontend).

## G Krav

- En fungerande applikation med tillhörande databas skriven i Razor Pages eller MVC.
- Lägga upp länk och .zip på PingPong.
- En kort text i README som beskriver ASP.NET Core och dess användningsområde
- En kort text i README som beskriver vilka delar som ingår i en Razor Pages applikation och kortfattat om hur de olika delarna hänger ihop.
- En kort text i README som beskriver vilka delar som ingår i en MVC applikation och kortfattat om hur de olika delarna hänger ihop.

## Deadline

**Deadline inlämningsuppgift (fre 26/3)**

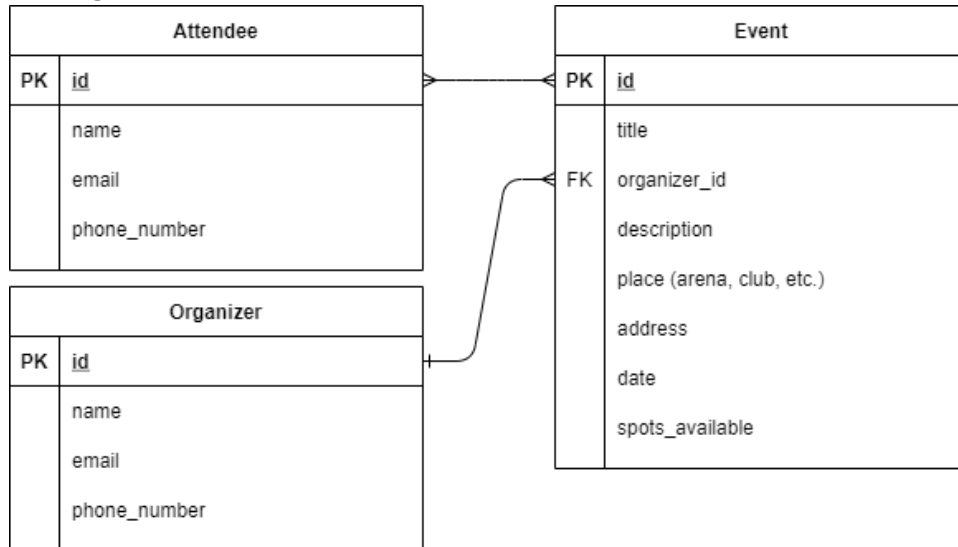
## Beskrivning

Webbapplikationen du skapar ska innehålla följande:

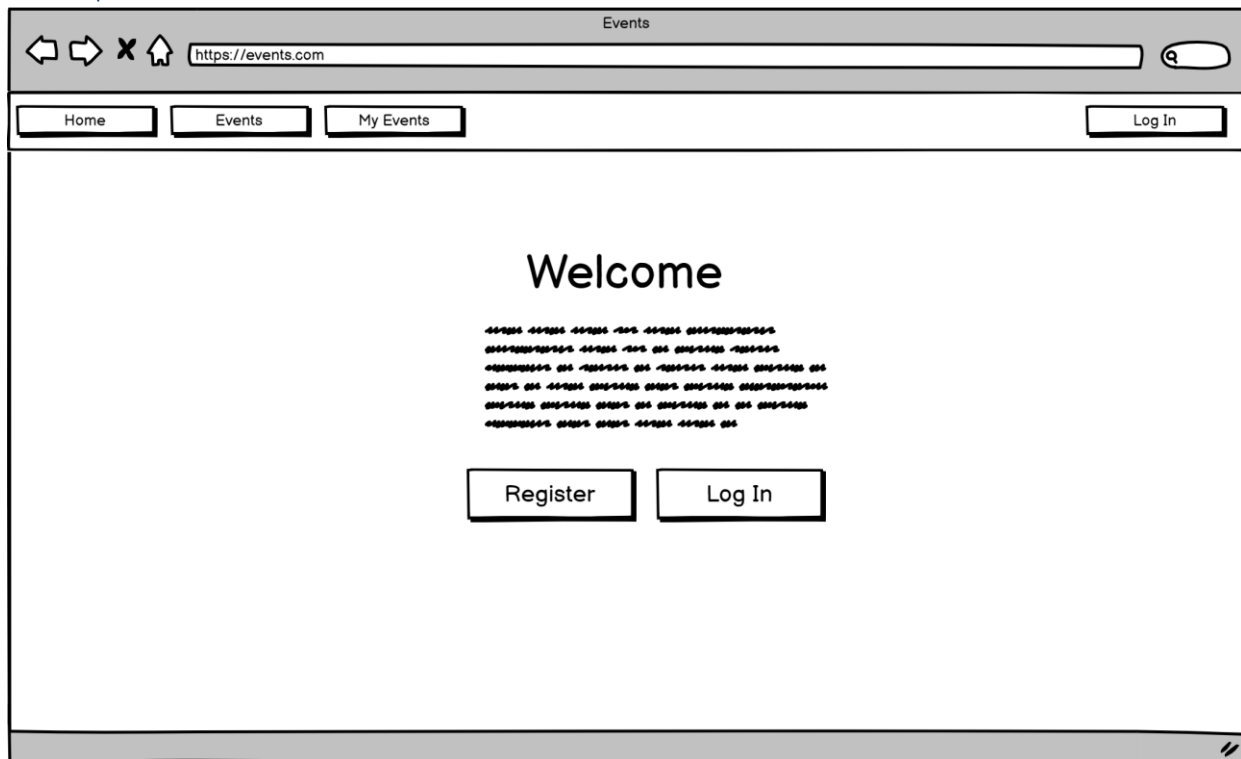
- Gemensam navigationsmeny med knapp för att gå till Home & Events & My Events sidorna. Samt en icke-funktionell "Log In" knapp som bara leder till Events.
- Home page: Förklarar lite vad sidan går ut på och erbjuder användaren att logga in/registrera via ett par knappar (båda leder direkt till Events sidan för vi har inte gått igenom autentisering&identifiering ännu).

- Events page: Här visas en aktuell lista över alla existerande event som finns i systemet. Varje event har en Join knapp som leder till Join Event sidan.
- My Events page: En lista över alla events som användaren har gått med i.
- Join Event page: En sida som visar detaljerad info om ett event och har en <form> tag för att gå med i eventet. Efter att man gått med i eventet så erbjuds man att gå till Events eller My Events sidorna.
- Eventdata och registreringsdata i en databas med hjälp av EF Core enligt ER-diagramm.

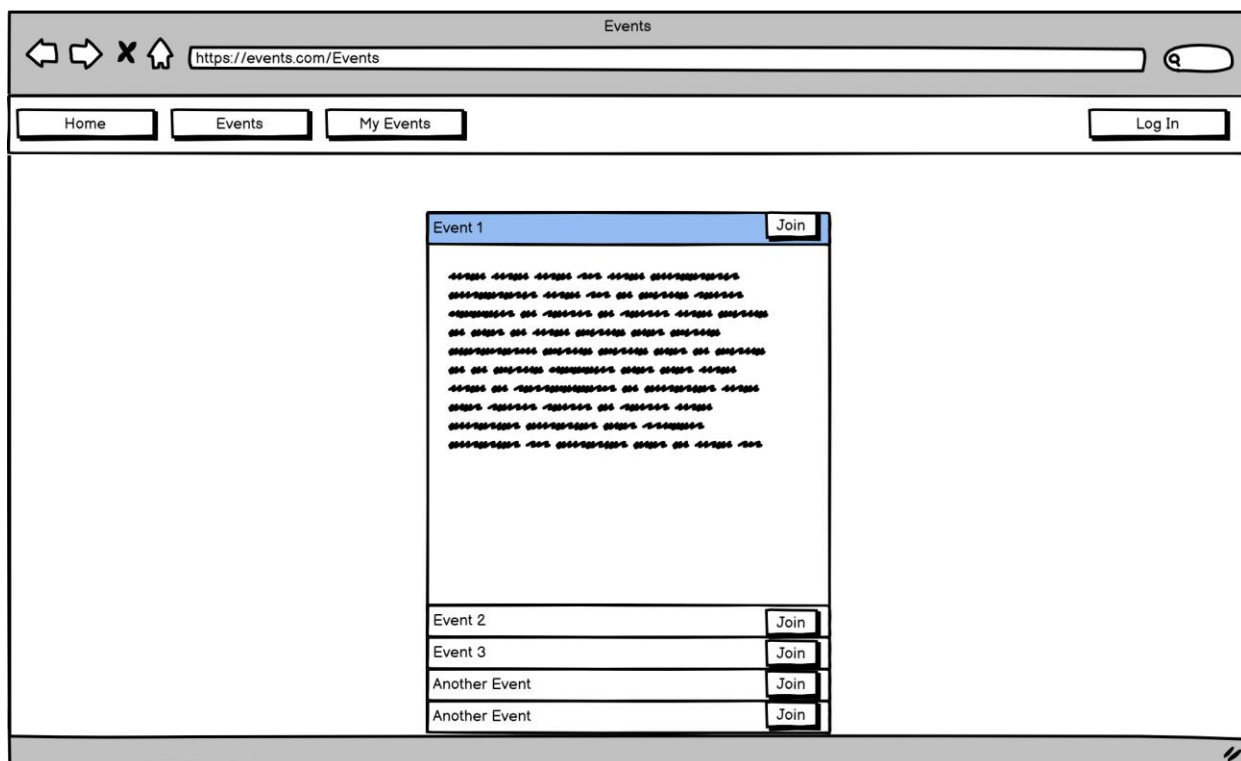
ER-diagramm:



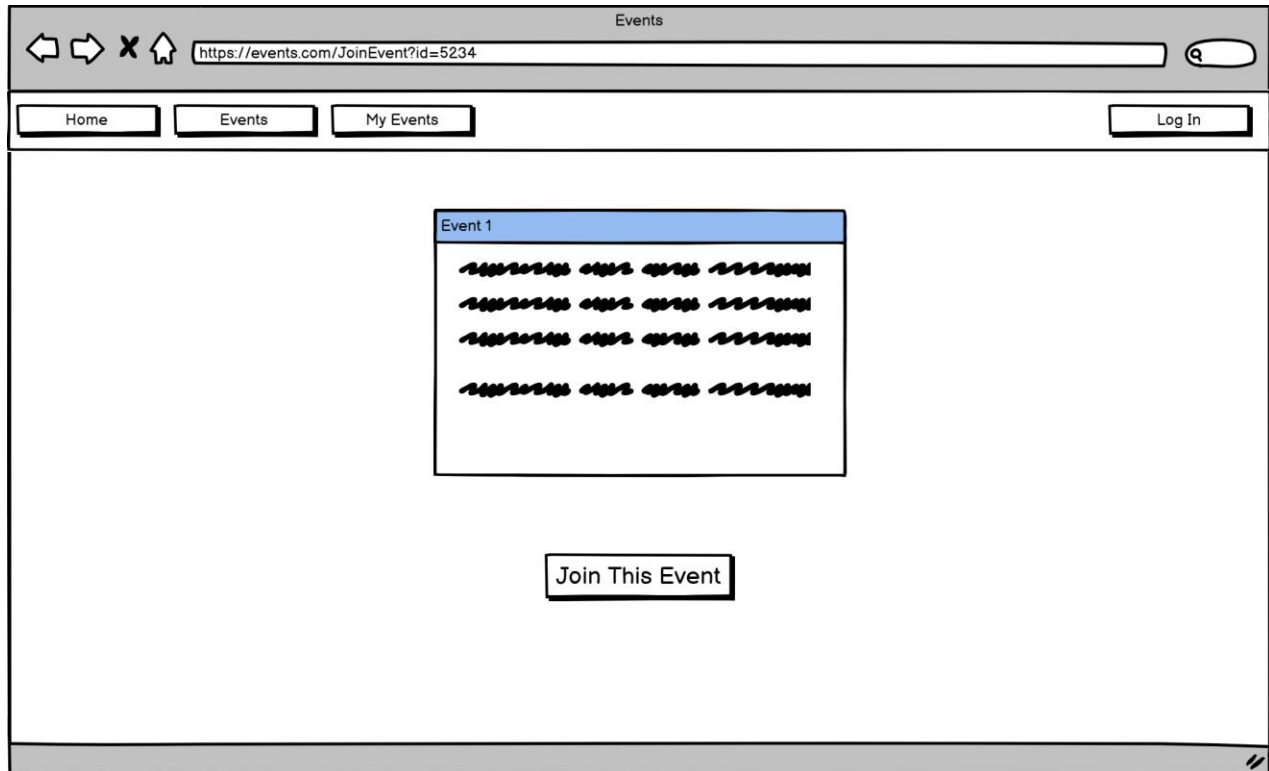
Mockup:



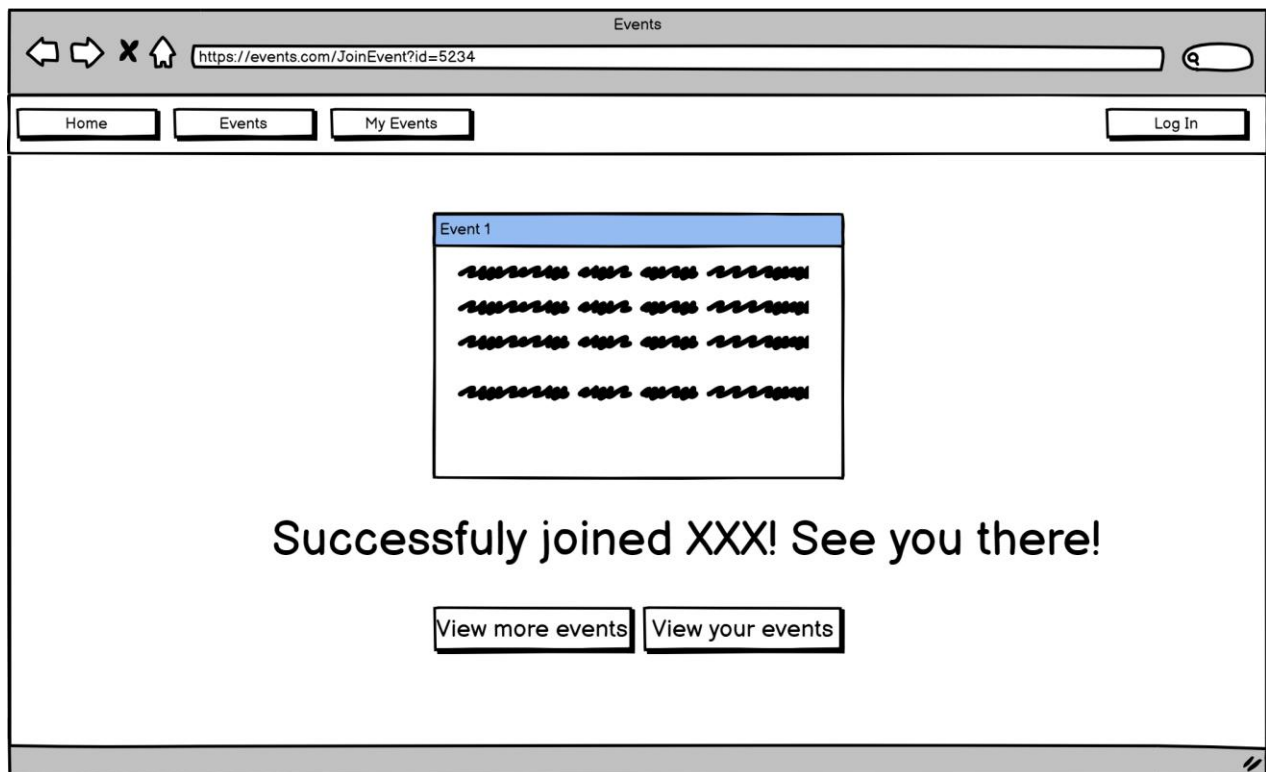
Registrering&loginknapparna kommer inte att användas i detta tillfälle, men senare när du övar på autentisering/identifiering så kommer du att aktivera dem. Nu länkar båda till Events sidan.



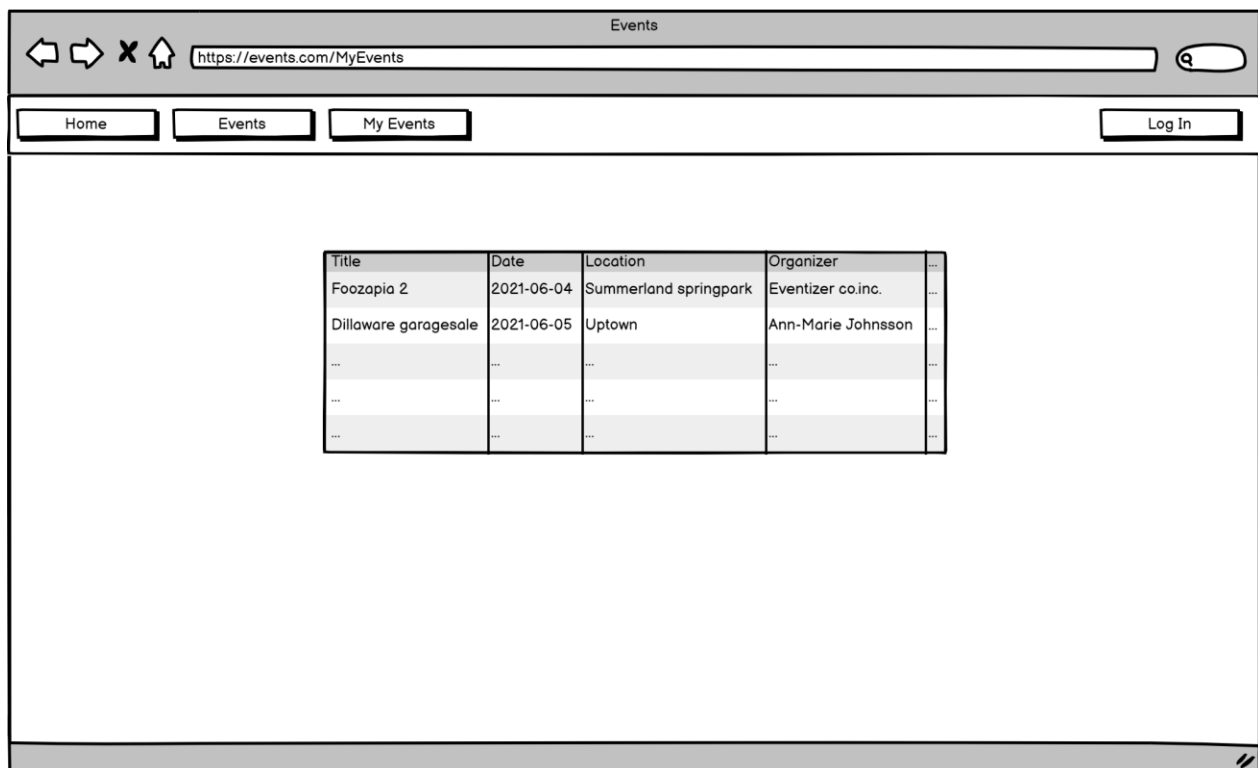
I exemplet används en dragspelsmeny (accordion) för att visa eventen, men här kan du göra på det sättet du är mest bekväm med. Det viktiga är att det går att se informationen om eventen från din databas och att du länkar till JoinEvent sidan med rätt Event ID som search parameter för varje Join knapp.



Efter att man har klickat på Join så kommer man hit. Notera URL:en. Detta är GET versionen av sidan.



POST versionen av sidan när man gått med i ett event.



En enkel tabulatur för att visa vilka events man har gått med i.

### Avgränsning:

I den här uppgiften ligger fokus på att skapa en fungerande applikation i Razor Pages, fokus ligger inte på att skapa en snygg frontend med JavaScript, JQuery, etc. Du får använda t.ex. Bootstrap men lägg inte fokus på det då det som bedöms i uppgiften är att du har skapat en webbapplikation med den beskrivna funktionaliteten i ASP.NET.

### GitHub:

Använd Git för att versionshantera koden. Committa varja dag och ofta. Ha din kod på ett publikt GitHub repo som du länkar till när du lämnar in din .zip av projektet. **Både länk och .zip på PingPong.**

### Punktlista

1. Skapa modellklasser för de tre databastables som behövs i en 'Models' folder.
2. NuGet nödvändiga EFCore bibliotek
3. Använd scaffolding(list) för att skapa Events sidan.
4. Se till att context klassen ser rätt ut och lägg till seeding. Inspektera databasen i VS eller SSMS.
5. Använd scaffolding för att lägga till My Events(list) och Join Event(details) sidorna.
6. Lägg till Index sidan och länka ihop alla sidorna.
7. Ta bort/modifiera den template kod du inte behöver i projektet (Privacy, Edit/Delete.. länkar i Events, etc.)
8. Lägg till en <form> på Join Event sidan och se till att uppdatera databasen på OnPost (använd alltid samma User i denna inlämningsuppgift)
9. Testa din sida och se att allt fungerar som det ska.
10. Lägg till README i projektet, läs G kraven noga.
  - För Asp.Net förklaringen är det fokus på det som sker i Startup.cs, wwwroot och Razor språket.
  - För Razor Pages så handlar det bara att beskriva de två filerna. Content Page och Page Model.
  - För MVC så är det skillnaden mellan Model, View och Controller.